Vision-Based Closed-Loop Tracking Using Micro Air Vehicles

Takuma Nakamura
Georgia Institute of Technology
270 Ferst Dr.
Atlanta, GA 30332
takuma.nakamura@gatech.edu

Stephen Haviland
Georgia Institute of Technology
270 Ferst Dr.
Atlanta, GA 30332
shaviland3@gatech.edu

Dmitry Bershadsky
Georgia Institute of Technology
270 Ferst Dr.
Atlanta, GA 30332
dbershadsky@gatech.edu

Daniel Magree
NodeIn
12 Deerfield Trace
Burlington, CT 06013
dmagree@gmail.com

Eric N. Johnson
Georgia Institute of Technology
270 Ferst Dr.
Atlanta, GA 30332
eric.johnson@ae.gatech.edu

***Abstract*—This paper describes the target detection and tracking architecture used by the Georgia Tech Aerial Robotics team for the American Helicopter Society (AHS) Micro Aerial Vehicle (MAV) challenge. The vision system described enables vision-aided navigation with additional abilities such as target detection and tracking all performed onboard the vehicles computer. The author suggests a robust target tracking method that does not solely depend on the image obtained from a camera, but also utilizes the other sensor outputs and runs a target location estimator. The machine learning based target identification method uses Haar-like classifiers to extract the target candidate points. The raw measurements are plugged into multiple Extended Kalman Filters (EKFs). The statistical test (Z-test) is used to bound the measurement, and solve the corresponding problem. Using Multiple EKFs allows us not only to optimally estimate the target location, but also to use the information as one of the criteria to evaluate the tracking performance. The MAV utilizes performance-based criteria that determine whether or not to initiate a maneuver such as hover or land over/on the target. The performance criteria are closed in the loop which allows the system to determine at any time whether or not to continue with the maneuver. For Vision-aided Inertial Navigation System (V-INS), a corner Harris algorithm finds the feature points, and we track them using the statistical knowledge. The feature point locations are integrated in Bierman Thornton extended Kalman Filter (BTEKF) with Inertial Measurement Unit (IMU) and sonar sensor outputs to generate vehicle states: position, velocity, attitude, accelerometer and gyroscope biases. A 6-degrees-of-freedom quadrotor flight simulator is developed to test the suggested method. This paper provides the simulation results of the vision-based maneuvers: hovering over the target, and landing on the target. In addition to the simulation results, flight tests have been conducted to show and validate the system performance. The 500 gram Georgia Tech Quadrotor (GTQ)-Mini, was used for the flight tests. All processing is done onboard the vehicle and it is able to operate without human interaction. Both of the simulation and flight test results show the effectiveness of the suggested method. This system and vehicle were used for the AHS 2015 MAV Student Challenge where the GPS-denied closed-loop target search is required. The vehicle successfully found the ground target, and landed on the desired location. This paper shares the data obtained from the competition.**

TABLE OF CONTENTS

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have attracted much attention for many years because of their potential application in a wide singular of realms such as military, entertainment, and emergency responders. Electric MAV quadrotors are one type of UAVs that has emerged very recently thanks to the technology advancement in several areas: MEMS sensors, high energy density batteries, and high-performance microcontrollers to name a few. Although many of the UAVs had been mainly used for the outdoor missions, the size of the MAVs allows them access to indoor missions such as a disaster response to destroyed/bio-hazard buildings, serving people, and carrying luggage. To be able to operate in an indoor setting the vehicle must be capable of navigating without Global Positioning System (GPS), and be able to sense and avoid objects. The system to be described uses vision to supplement the loss of GPS indoors.

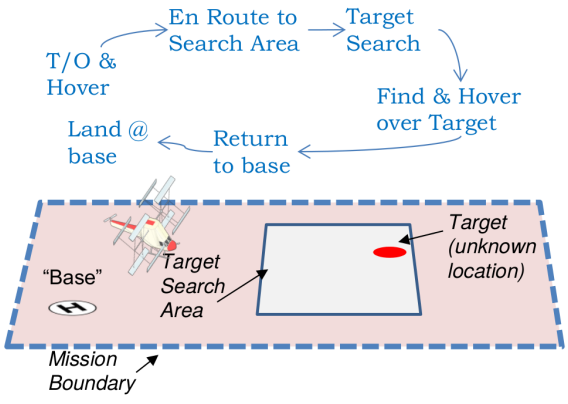


Figure 1. Schematic of mission in 2015 AHS MAV Student Challenge.

In this paper, the authors propose a target detection and tracking method using a self-contained MAV equipped with

a monocular camera, IMU, magnetometer, and sonar. The overview of the object mission is shown in Figure 1, which is cited from the official competition rules of 2015 AHS MAV Student Challenge[2]. In this challenge, a MAV autonomously takes off from a helipad and using no external aides including GPS to find a target in a search area. Once found, the vehicle hovers over the target, and returns to land on the helipad. The exact target location is unknown, we have a priori visual information of the target, operation time is strictly limited, and the vehicle size and weight are restricted. There are two significant factors for achieving the mission: real time localization and target recognition.

When MAVs are limited in size and weight there are benefits such as access to smaller, harder to reach places and ease on operators but disadvantages are payload and flight time. In [22], [21], [7] [13], a stereo camera based localization approach is presented. The benefit of stereo camera is riddance of ambiguity in depth that a monocular camera has, and this results in accurate localization. As a trade-off, this approach imposes a higher processing burden and sensitive calibration of two cameras. Also, the weight of the vehicle increases, and duration decreases. Many researchers showed the feasibility of single monocular camera based localization [10], [9]. Since the duration is critical for this mission, and processing power also needs to be used for the target detection, a monocular camera approach is chosen in this development.

In recent years computer vision literature has contained considerable information on a target detection. The achievement of the line detection by a Hough transform [2] has led to a more robust methodology[17]. A considerable amount of literature deals with the feature point detection.[5] There exists much work performed to detect a known target using machine learning [12], [25], [24]. To extend the application of target detection, attempts have been made to track the optical flow using sequential images rather than one independent measurement [14], [8], and built a solid foundation for visual servoing [18]. However, most of the work has been concentrated on the method merely using a camera as a independent sensor, and few of the work has integrated with other sensors on UAVs [16], [26], [27]. Our work uses a different method for target recognition, and runs separated estimators for the vehicle pose and the target position. Our paper provides considerable insight into the combination UAVs and Haar-like feature detector using multiple EKFs.

In this paper, we develop a self-contained fully autonomous vision-based MAV quadrotor for indoor search missions. This work is an extension of the results reported in our previous paper [6]; therefore, some of the details addressing the hardware design and software implementation have been omitted. In Section 2, we describe our vision system to detect and track the target. Section 3 explores the vehicle and sensor models including the state estimation using EKF. In Section 4, the simulation results for vision-based hovering and landing are provided. Also, this section gives the simulation results with the recorded data in flight test using 500-gram MAV. This data allows us to tune the parameters in the EKFs. Finally, Section 5 describes the full scale mission results. The full-scale mission is operated under the rule of 2015 AHS MAV student challenge.

[2]**https://vtol.org/71D5B0C0-58CC-11E4-9D190050568D0042**

2. TARGET RECOGNITION

The target pattern is detected using a machine learning based algorithm called a Haar-like feature detector. The raw measurements outputted from the Haar detector is tracked by multiple EKFs, and their expected error covariance matrices provide us with the uncertainty of each of the EKFs. The uncertainty is used as one of the criteria to evaluate the performance of the filter as well as the convergence criteria which needs to be satisfied before the estimation is used for navigation. Z-test allows the filter to solve the corresponding problem; that is, to decide which measurement should be plugged into which EKF for updating the state.

Haar-like feature detector

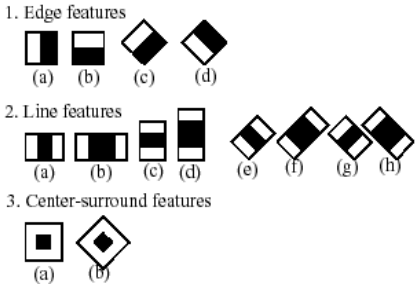


Figure 2. Example of Haar-like features.

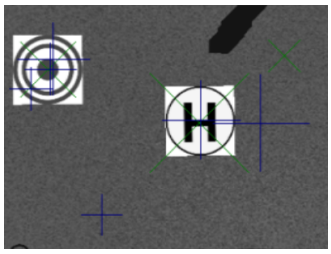


Figure 3. Example scenes with Haar-like feature detectors. The horizontal and vertical crossings are the measurement output from the helipad classifier, and the diagonal crossings are from the double-red-circle classifier. The length of the crossing corresponds to the estimated size of the target.

The Haar-like feature detector is one type of feature-based approach to object classification. Typically, a feature-based approach functions operate faster than the pixel based [25]; therefore, it is suitable for real-time applications such as target tracking using UAVs. We show in Figure 2 some of the basic features. The Gentle Adaptive Boost Algorithm (Ada Boost) uses these simple features to train a classifier [3]. One classifier was trained for each of the targets in the mission, i.e., double-red circle and helipad. Ada Boost needs images that contain the object of interest, which are so called positive images, and images that do not contain the object of interest, which are so called negative images. We used 50 positive images and 600 negative images to train each of the classifiers. The raw outputs of the Haar-like feature detector are shown in the Figure 3. As shown in the figure, the raw measurements are noisy, and often mis-understand that the helipad is a double-red-circle, and vice versa. Also, the estimated sizes of the targets deviate from the actual sizes. The target size $(a, b[\text{pixel}])$ in camera can be estimated using the vehicle altitude as follows:

$$a = -\frac{\text{WidthA}}{2^b\hat{Z}_i^i \tan(\frac{\gamma_x}{2})}, \quad b = -\frac{\text{WidthB}}{2^b\hat{Z}_i^i \tan(\frac{\gamma_y}{2})}, \tag{1}$$

where A, B(ft) are the dimensions of the target, γ_x, γ_y are the horizontal and vertical field of view, respectively and ${}^b\hat{Z}_i^i$ is the z-direction position of the vehicle (b) with respect to the origin of the inertial frame (superscript i) expressed in inertial frame (subscript i). The same notation is used throughout this paper. The classifier only searches the features with the estimated size. In order to allow the estimation error in target size, we give k-pixel margin, i.e., the detection starts with the smallest classifier size $a-k, b-k$ and ends with $a+k, b+k$. This technique not only reduces the wrong positive hit, but also speeds up the process. The k factor is tuned in Section 4 using the flight test data.

Propagation of EKF

The measurement of the Haar-like feature detector is expressed as:

$$z_i = \begin{bmatrix} u \\ v \end{bmatrix}_i + v_k = \begin{bmatrix} f_x \frac{{}^t\hat{Y}_c^c}{{}^t\hat{X}_c^c} \\ f_y \frac{{}^t\hat{Z}_c^c}{{}^t\hat{X}_c^c} \end{bmatrix} + v_k, \tag{2}$$

where i is the index for the measurements at the same time, w_k is process noise, and v_k is measurement noise. Both of them are zero-mean Gaussian, i.e. $w_k \sim N(0, Q_k)$ and $v_k \sim N(0, R_k)$. Up to 20 raw measurements are stored for the Z-test, and the best measurements are plugged into the Kalman filter update. The state of the estimator, $\hat{x}$, is the position of the target in inertial frame, and the z position is overwritten by the output of the vehicle state estimation. Now, let us express the dynamics in state space.

$$x_{k+1} = Fx_k + w_k, \tag{3}$$

$$y_k = H_k x_k + v_k, \tag{4}$$

Note that since the target is static in this mission, $F = I_3$. The output matrix H cannot be directly computed, but using the position of the target expressed in the camera frame [1] it can be expressed as:

$$\begin{aligned} H &= \frac{\partial z}{\partial r_i}|_{x=\hat{x}} \\ &= \frac{\partial z}{\partial r_c}\frac{\partial r_c}{\partial r_i}|_{x=\hat{x}} \\ &= \begin{bmatrix} \frac{\partial u}{\partial {}^t\hat{X}_c^c} & \frac{\partial u}{\partial {}^t\hat{Y}_c^c} & \frac{\partial u}{\partial {}^t\hat{Z}_c^c} \\ \frac{\partial v}{\partial {}^t\hat{X}_c^c} & \frac{\partial v}{\partial {}^t\hat{Y}_c^c} & \frac{\partial v}{\partial {}^t\hat{Z}_c^c} \end{bmatrix} L_{ci} \\ &= \begin{bmatrix} -f_x \frac{{}^t\hat{Y}_c^c}{{}^t\hat{X}_c^c} & f_x \frac{1}{{}^t\hat{X}_c^c} & 0 \\ -f_y \frac{{}^t\hat{Z}_c^c}{{}^t\hat{X}_c^c} & 0 & f_y \frac{1}{{}^t\hat{X}_c^c} \end{bmatrix} L_{ci}, \end{aligned} \tag{5}$$

where f_x, f_y is the focal length of the camera, and the rotation matrix from camera to inertial is denoted $L_{ic} = L_{ci}^T$. Using the camera tilt (α), and pan (β) angle which are defined with a right-handed coordinate system with x-axis being the optical axis, the transformation matrix from body to camera frame can be expressed as follows:

$$L_{cb} = \begin{bmatrix} cos(\alpha) & 0 & -sin(\alpha) \\ sin(\beta)sin(\alpha) & cos(\beta) & sin(\beta)cos(\alpha) \\ cos(\beta)sin(\alpha) & -sin(\beta) & cos(\beta)cos(\beta) \end{bmatrix}, \tag{6}$$

The transformation matrix from inertial to camera frame is obtained as

$$L_{ci} = L_{cb}L_{bi}. \tag{7}$$

Note that L_{bi} can be expressed in various ways, and the Section 3 describes the method used in our development. The focal length f_x, f_y is expressed as follows using the width and height of the image:

$$f_x = \frac{\text{Width}}{2tan(\frac{\gamma_x}{2})}, \quad f_y = \frac{\text{Height}}{2tan(\frac{\gamma_y}{2})}, \tag{8}$$

In the propagation of the EKF estimation, the covariance matrix, P, is updated using the discrete Lyapunov equation as follows:

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1}. \tag{9}$$

Since the target is known to be static, the state of the estimator, $\hat{x}$, is replaced by the previous posterior state, i.e. $\hat{x}_k^- = x_{k-1}^+$.The propagation of the error covariance matrix is expressed as $P_k^- = P_{k-1}^+ + Q_{k-1}$.

Update of EKF

In the update phase, the state and covariance matrix are updated using the Kalman gain matrix, which is obtained from the below equation:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}. \tag{10}$$

Although the linear model is available for the update of the state, the non-linear function, $h(x)$ that takes in the current state and computes an expected measurement vector is used for the update to reduce the issues related linearization.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k)), \tag{11}$$

$$P_k^+ = (I_3 - K_k H_k)P_k^-, \tag{12}$$

Z-Test

The statistical test, which is also called Z-test, is used to solve the corresponding problem. Since there are multiple outputs from the Haar-like feature detector, we need to choose an output used for the measurement update. The Z value, also called Mahalanobis distance, is defined as follows:

$$Z = e_k^T (H_k P_k H_k^T + R_k)^{-1} e_k, \tag{13}$$

where e_k is the residual. The measurement with the smallest Mahalanobis distance is used for the EKF update. The constant threshold for the Z value is set. This way, when the estimation is less confident, the measurement with a large residual is allowed for the update, and vice versa.

Initialization

The EKF is only initialized after observing consistent measurements. Re-initialization occurs either when

1. the determinant of P exceeds a certain value, (this corresponds to the case when the estimator is very unconfident) or
2. consistent unavailability of the measurement that satisfies the z-test.

The diagonal entries of P are set to large numbers, and the off-diagonal numbers are zeros. The initial state of the EKF is the average of the states observed during the initialization process.

Convergence Criteria

Only when the current estimate is considered to be accurate to some extent, the estimated state is used for navigation. There are 2 criteria: both the determinant of P and absolute change in the state have to be smaller than a certain value, which can be expressed as follows:

$$\det(\mathrm{P}) \leq \varepsilon_{\mathrm{P}}, \tag{14}$$

$$|\hat{x}_k - \hat{x}_{k-1}| \leq \varepsilon_x, \tag{15}$$

where ε_P and ε_x is the design parameter to update the desired waypoint based on determinant and absolute position change, respectively. When multiple EKFs satisfies the criteria, the EKF with minimum $\det(\mathrm{P})$ is used for navigation.

3. MODELING

This section describes the flight simulator developed to test the target tracking method explained in the Section 2. A 6-degrees-of-freedom quadcopter simulator is developed with a simulated on-board camera. It uses a PD control for attitude control and PID controller for position control. The state estimation is achieved with the hybrid extended Kalman filter, i.e., the Kalman filter prediction uses continuous vehicle dynamics, and its update is computed using discrete systems.

Process Model

In this flight simulator, the simplest model that is sufficient to test the target tracking algorithm is chosen. The kinematics only consider the inertial and vehicle body frames, and ignores the rotation of the Earth and other complex models. The center of gravity of the vehicle is assumed to be located at the origin of the body frame. The state vector of the simulation process model has 13 states as follows:

$$\vec{x} = \begin{bmatrix} {}^b p_i^i & {}^b V_b^i & {}^b q^i & {}^b \omega^i \end{bmatrix}^T, \tag{16}$$

where q is an attitude quaternion. Let ΣF and ΣG denote the total force and moment applied to the center of mass of the vehicle. The equation of motion can be written as follows:

$$\Sigma F_b = m \frac{d^b V_b^i}{dt} + m({}^b \omega^i \times {}^b V_b^i), \tag{17}$$

$$\Sigma G_b = I_b \frac{d^b \omega^i}{dt} + \omega \times (I_b \, {}^b \omega^i), \tag{18}$$

where I_b is an body inertia matrix expressed in the body frame, and m is the body mass. These two equations give the time derivative of axial and angular velocity of the body. The time derivative of the position is obtained as follows:

$$\frac{d^b p_i^b}{dt} = L_{ib} \, {}^b V_b^i, \tag{19}$$

where $L_{bi} = L_{ib}^T$ is a rotation matrix from inertial to body frame defined as follows:

$$L_{bi}(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{20}$$

Finally, the attitude quaternion is updated using the following equation:

$$\frac{d^b q^i}{dt} = \frac{1}{2} \mathbf{Q}({}^b \omega^i) {}^b q^i, \tag{21}$$

where the function $\mathbf{Q}(\omega) : \mathbb{R}^{3\times 1} \to \mathbb{R}^{4\times 4}$ is defined as follows:

$$\mathbf{Q}(\begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix}) = \begin{bmatrix} 0 & -\omega_0 & -\omega_1 & -\omega_2 \\ \omega_0 & 0 & \omega_2 & -\omega_1 \\ \omega_1 & -\omega_2 & 0 & \omega_0 \\ \omega_2 & \omega_1 & -\omega_0 & 0 \end{bmatrix}. \tag{22}$$

Actuator Model

Each of the four motors of this model produces the force and moments proportional to the square of angular velocity, ω_i, which is expressed as:

$$F_i = k_F \omega_i^2, \quad M_i = k_M \omega_i^2, \tag{23}$$

where k_F and k_M is a constant value decided by the experiment. Although typical quadrotors use three-phase brushless motor, the dynamics can be approximated by a first-order differential equation of a DC motor:

$$\dot{\omega}_i = k_m(\omega_i^d - \omega_i), \tag{24}$$

where ω_i^d is the desired angular velocity determined by the attitude controller. The actuator rotation per minute (RPM) is bounded, and the control values are given as a RPM. In hover, the generated force is equal to the gravity. Therefore, the force generated by each rotor in hover ($F_{i,h}$) is expressed as follows:

$$F_{i,h} = \frac{mg}{4}. \tag{25}$$

Finally, the required RPM in hover is

$$\omega_h = \sqrt{\frac{mg}{4k_F}}. \tag{26}$$

Sensor Models

The vehicle is equipped with an IMU, sonar, magnetometer, and camera. Each of the sensors has a noise and bias. In this paper, the accelerometer and gyroscope is corrupted by noise and bias as follows:

$$s_{raw} = s_{true} + {}^s b + L_{bi} g + \xi_s, \tag{27}$$
$$\omega_{raw} = \omega_{true} + {}^\omega b + \xi_\omega, \tag{28}$$

where g is a gravitational acceleration vector expressed in an inertial frame, i.e. $g = \begin{bmatrix} 0 & 0 & 32.174[ft/s^2] \end{bmatrix}^T$, and ξ is a zero-mean Gaussian noise. ($\xi_s = N(0, \sigma_s^2)$ and $\xi_\omega = N(0, \sigma_\omega^2)$). In the simulation, IMU data is available at each time step: a postulate of the usage of a continuous Kalman filter for propagation. The simulation runs at 1000Hz, and thus the IMU sampling time is 0.001s. The measurement of sonar and magnetometer is assumed to be corrupted only by noise,

$$Sonar_{raw} = {}^b Z_i + \xi_{sonar}, \tag{29}$$
$$Mag_{raw} = {}^b \psi_i + \xi_{mag}. \tag{30}$$

Estimation

EKF propagation

The vehicle estimator has the state of

$${}^b\hat{x} = \begin{bmatrix} \hat{p}_i & \hat{v}_b & \hat{q}_i & {}^s\hat{b} & {}^\omega\hat{b} \end{bmatrix}^T \tag{31}$$

where p, v, q is the position, velocity, attitude quaternion respectively, and ${}^s b, {}^\omega b$ is the bias of the accelerometer and gyroscope, respectively. The quaternion is used to avoid the singularity that can be caused by Euler angles, but it still can cause a singularity of the covariance matrix in the EKF [28]. This singularity problem is solved by defining an infinitesimal error quaternion δq [11] as follows:

$$\delta q = \begin{bmatrix} 1 & r \end{bmatrix}^T. \tag{32}$$

The $r \in \mathbb{R}^{3\times 1}$ is tracked as a minimal representation of the attitude state error to the reference state. The quaternion is propagated using

$$\delta q = q_{ref}^{-1} * \hat{q}, \tag{33}$$

where q_{ref} is a reference attitude, which is zero-tilting attitude in hover. Note that "$*$" indicates quaternion multiplication. Provided two quaternions as follows:

$$p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} p_0 \\ \vec{p} \end{bmatrix}, q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \tag{34}$$

Then, the quaternion multiplication is defined as follows.

$$\begin{aligned} p * q &= \begin{bmatrix} p_0 q_0 - \vec{p} \cdot \vec{q} \\ p_0 \vec{q} + q_0 \vec{p} + \vec{p} \times \vec{q} \end{bmatrix} \\ &= \begin{bmatrix} p_0 q_0 - (p_1 q_1 + p_2 q_2 + p_3 q_3) \\ \begin{bmatrix} p_0 q_1 \\ p_0 q_2 \\ p_0 q_3 \end{bmatrix} + \begin{bmatrix} q_0 p_1 \\ q_0 p_2 \\ q_0 p_3 \end{bmatrix} + \begin{bmatrix} p_2 q_3 - p_3 q_2 \\ p_3 q_1 - p_1 q_3 \\ p_1 q_2 - p_2 q_1 \end{bmatrix} \end{bmatrix}. \end{aligned} \tag{35}$$

The details of the quaternion properties are described in this textbook [23]. The 16-element states defined in the Equation (31) is reduced to the 15-element state vector as below:

$${}^b\hat{x} = \begin{bmatrix} \hat{r} & \hat{p}_i & \hat{v}_b & {}^s\hat{b} & {}^\omega\hat{b} \end{bmatrix}^T \tag{36}$$

The state propagation is achieved using the full non-linear 6-degrees-of-freedom equation of motion as follows:

$$\dot{\hat{q}} = \frac{1}{2}\mathbf{Q}(\omega_{raw} - {}^\omega\hat{b}) \tag{37}$$

$$\dot{\hat{p}}_i = L_{ib}(\hat{v}_b), \tag{38}$$

$$\dot{\hat{v}}_b = s_{raw} - {}^s\hat{b}, \tag{39}$$

$${}^s\dot{\hat{b}} = 0, \tag{40}$$
$${}^\omega\dot{\hat{b}} = 0. \tag{41}$$

Note that $\mathbf{Q}$ is defined in the Equation (22). Although the state is propagated with non-linear equations, the error covariance matrix propagation requires the linear system because it is propagated with the continuous Lyapunov equation:

$$\dot{P} = AP + PA^T + Q, \tag{42}$$

where Q is process noise covariance matrix, and A is a state-transition matrix, respectively. In this case, process noise is the measurement noise of accelerometer and gyroscope. More details of the EKF with this state vector are described here [19].

EKF update

Although the propagation is conducted using continuous time, the other sensor outputs are relatively slow in terms of frequency, and they are updated with the discrete system. Since the sampling time of each sensor is different, the sensor update is processed sequentially as any of the sensors has a new measurement. The Kalman gain is obtained by the Equations (10), (11), (12). Note that H matrix varies for each of the sensors. The H matrix for each sensors is expressed as follows:

$$H_{Vicon} = \left[\begin{array}{c|c|c|c} & I_3 & & \\ \hline & L_{bi}(\hat{q}) & & \end{array}\right], \tag{43}$$

$$H_{sonar} = [\ \ |\ [0 \quad 0 \quad 1]\ |\ \ |\ \ |\ \ ] \tag{44}$$

$$H_{mag} = [\ L_{ib}(\hat{q})_3\ |\ \ |\ \ |\ \ |\ \ ], \tag{45}$$

where $L_{ib}(\hat{q})_3$ means the 3rd row of the transformation matrix. Note that the matrix is sectioned for a corresponding state vector (Equation (36)).

Control

Position Control—For position control, a PID controller is implemented. It is used for the vision-based hover and landing phase. The position controller takes a desired waypoint as a reference value, $\vec{r}^d = \begin{bmatrix} X^d & Y^d & Z^d \end{bmatrix}^T \in \mathbb{R}^3$ and is formalized as,

$$\begin{aligned} \ddot{\vec{r}} &= k_p(\vec{r}^d - \vec{r}) - k_d\dot{\vec{r}} + k_i \int (\vec{r}^d - \vec{r}) \\ &= k_p e_p + k_d e_d + k_i e_i \quad (\because \dot{\vec{r}}^d = \ddot{\vec{r}}^d = 0 \ \text{ in hover}), \end{aligned} \tag{46}$$

which can be linearized as the following,

$$\ddot{r}_1^d = g(\theta^d cos\psi + \phi^d sin\psi), \tag{47}$$

$$\ddot{r}_2^d = g(\theta^d sin\psi - \phi^d cos\psi), \tag{48}$$

$$\ddot{r}_3^d = \frac{8k_F\omega_h}{m}\Delta\omega_F, \tag{49}$$

where ψ denotes the current vehicle heading. Inverting this gives us the desired throttle command and attitude that makes the computed acceleration from the controller. The throttle command is directly used as a command to change the RPM, and the desired attitude is used in the attitude control to generate a desired aileron and elevator command.

$$\phi^d = -\frac{1}{g}(\ddot{r}_1^d sin\psi + \ddot{r}_2^d cos\psi), \tag{50}$$

$$\theta^d = \frac{1}{g}(\ddot{r}_1^d cos\psi + \ddot{r}_2^d sin\psi), \tag{51}$$

$$\Delta\omega_F = -\frac{m}{8k_F\omega_h}\ddot{r}_3^d. \tag{52}$$

Attitude Control— Attitude controller takes in the desired attitude $(\phi^d, \theta^d, \psi^d)$ as reference values, and uses the estimated states as feedback. The ϕ^d, θ^d are calculated using the desired acceleration as explained in the previous subsections, and ψ^d is the direct design parameter.

$$\Delta\omega_\phi = k_{p,\phi}(\phi^d - \phi) + k_{d,\phi}(p^d - p), \tag{53}$$

$$\Delta\omega_\theta = k_{p,\theta}(\theta^d - \theta) + k_{d,\theta}(q^d - q), \tag{54}$$

$$\Delta\omega_\psi = k_{p,\psi}(\psi^d - \psi) + k_{d,\psi}(r^d - r). \tag{55}$$

Finally, each control input is converted to the RPM as follows:

$$\begin{bmatrix} \omega_1^d \\ \omega_2^d \\ \omega_3^d \\ \omega_4^d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_F \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix} \tag{56}$$

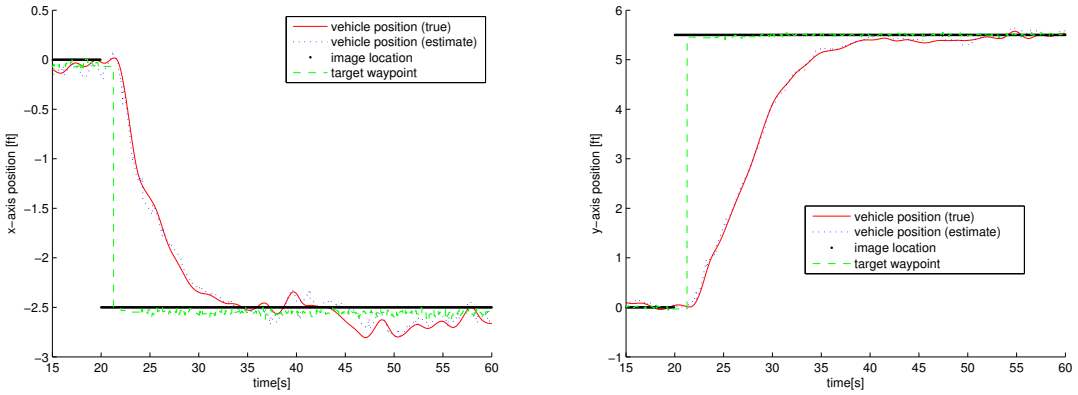
The details of the controller systems are described here [20].

4. SIMULATION RESULT

This section provides two types of simulation results. First, we show the simulation results of the intended mission maneuver: vision-based hovering and landing to validate our vision system and methodology. Second, we provide the simulation result with the recorded flight test. The purpose of this part is to tune several parameters using actual vehicle state and images obtained in flight tests.

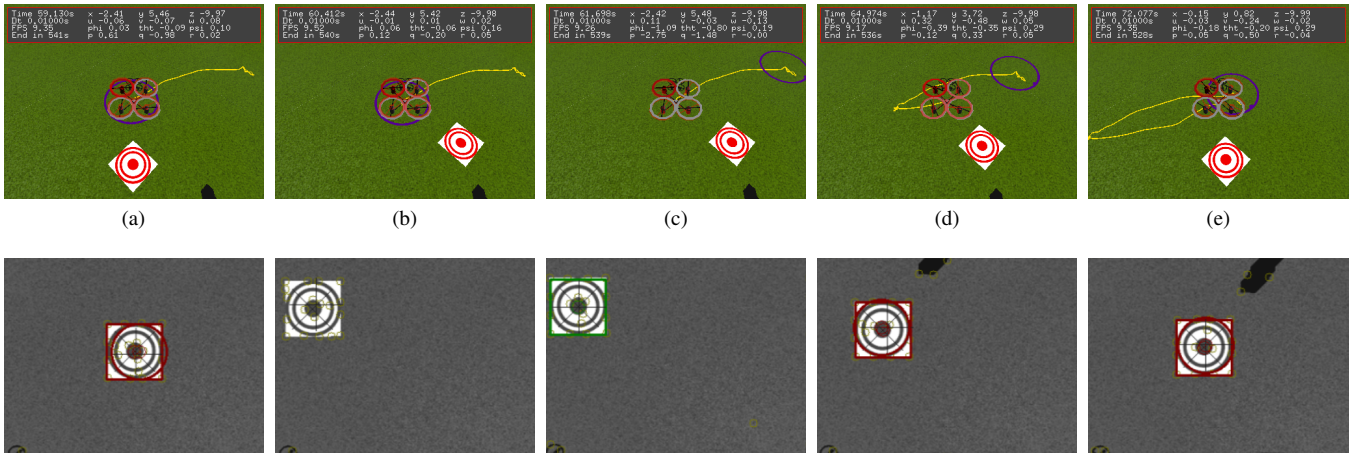
Vision-Based Hovering and Landing

In the simulation and experiment, the tilt angle was set to $\alpha = -90$ [deg], and the pan angle was set to $\beta = 180$ [deg] to make the onboard camera face down. The Haar-like feature detector processes the image rendered in the simulation. The desired position is the output of the Haar-like feature detector, and the position control uses EKF state estimation as feedback. Figure 4 shows the hovering result and the step response to the instantaneous change of the target image location. In this simulation, the target image location is moved at 20[s] in the simulation time. As shown in the figure, the target tracker does not immediately follow the new image location because that abrupt change of the output was understood as a noise by the estimator, and was eliminated in the Z-test. After consecutive measurements, one of the EKFs sees many consistent measurements around the new location, and the target waypoint shifts to the new point. Figure 5 shows this simulation. This result validates the accurate vision-based hovering. (The Root Mean Square Error (RMSE) is summarized in the Table 1.) Also, this methodology allows the vehicle to recover from disturbances as large as 6 ft when flying at 10 ft.The result of the landing simulation is summarized in Figure 6. The landing operation was initiated at 76[s] in the simulation. The vehicle successfully landed near the target. The RMSE of the operation is summarized in table 1. As expected, the target had not been updated after 89[s]. This is because it got no new measurements at some point of the descent, and the predicted error covariance increased and no longer satisfied the convergence criteria described in the Section 2. The desired vehicle position remained the last confident waypoint.



(a)x-direction (b)y-direction

Figure 4. Simulation of vision-based hovering. Step disturbance is given at 20[s] as an instantaneous change of the image location. The target tracker does not immediately move to the new target location. After consistent measurements, the target waypoint location is renewed, and the vehicle follows that signal.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5. Simulation of vision-based hovering with graphics. The upper figures show the vehicle motion, and the lower figures show the simulated onboard images. Initially, the vehicle was hovering over the target (a),(f). The target tracker lost the target after disturbance (b), (g), and regained the target after observing consistent measurements (c), (h). The vehicle moved toward the new location (d), (i) and backed to the hover status (e), (j).

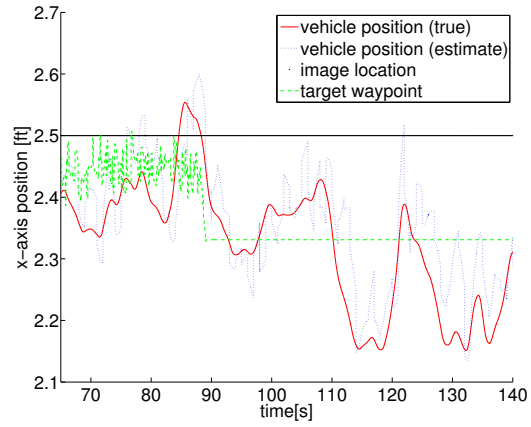
Table 1. **Summary of Simulation**

Maneuver	RMSE
Hovering (before disturbance)	0.1125[ft]
Hovering (after disturbance)	0.1090[ft]
Landing (average)	0.1296[ft]
Landing (terminal)	0.2173[ft]

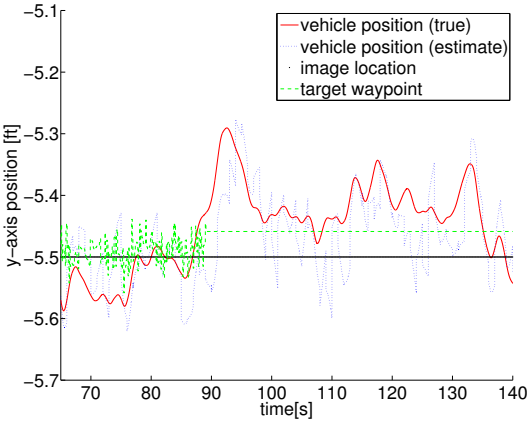
Simulation with the recorded data

The GTQ-Mini (Figure 8 (a)), was developed for the AHS MAV Student Challenge, and this vehicle was used for flight tests. The GTQ-Mini weighs under 500 grams and is less than 450mm in length in any dimension to satisfy the requirement of the student challenge. It is able to fly without any external aid such as computing power and Vicon cameras. Test flights were operated at the Georgia Tech Indoor Flight Facility (Figure 8 (b)) using Vicon systems, but this is only to obtain the actual target position and vehicle states to compare them with the estimated. This vehicle was developed with the UAV Research Facility's, UAVRF, Electronic Multirotor Sizing Tool (EMST) optimizer [6], which allowed the vehicle to achieve the desired operation time and weight. The onboard computer, Gigabyte Brix[3], runs on Ubuntu 14.04 Operation System, and the computer communicates via USB interface with an Ardupilot[4]. The Ardupilot provides inertial measurement unit (IMU) sensor readings to the Gigabyte Brix for processing. Custom USB cables were made to allow a MB1040 LV MAXSonar EZ4[5] sonar sensor and a downward facing Firefly-MV[6] USB monocular camera to connect to the onboard computer. Many sequential images were obtained while the vehicle was hovering on/ moving around the targets. The obtained images and Vicon data were plugged into the simulation to replicate the flight. Measurement noise, process

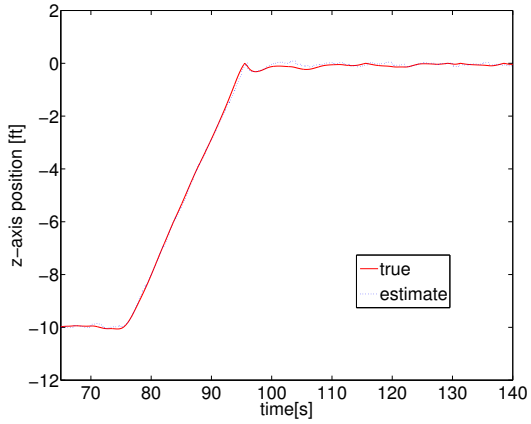
[3]**http://www.gigabyte.us/**
[4]**http://3drobotics.com/**
[5]**http://www.maxbotix.com**
[6]**http://www.ptgrey.com/**



(a)x-direction



(b)y-direction



(c)z-direction

Figure 6. Simulation of vision-based landing. The target image is static, and located at (2.5, -5.5). Target waypoint is the output of the target tracker, and vehicle tries to follow this position. The controller only knows the estimated vehicle position output from the EKF.

Table 2. Summary of Tuned Parameters

Parameter	Value
Process Noise Variance	$0.5[\text{ft}^2]$
Measurement Noise Variance	$8[pixel^2]$
k-factor	9[pixel]

noise, and k-factor is tuned to best perform the tracking. Figures 7 and 9 show the result of the flight test after tuning. The tuned parameters are summarized in Table 2. The results show that the average estimation error is less than 0.5 ft, and both of the double red circle, and helipad searching have the bias from the Vicon outputs. These biases can be explained by several factors: the poorly cropped images used for the training of the classifier, an improperly mounted onboard camera, Vicon measurement bias and so on. Overall, the suggested tracking method can be inferred to be accurate enough to be used for the hovering and landing.

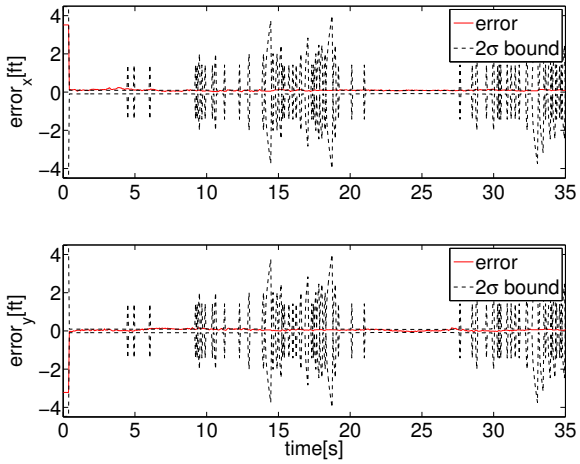


Figure 7. The plots show the error and two sigma bound of the helipad searching result after tuning. The errors are computed using the Vicon outputs. The position estimation is slightly biased to the positive direction.

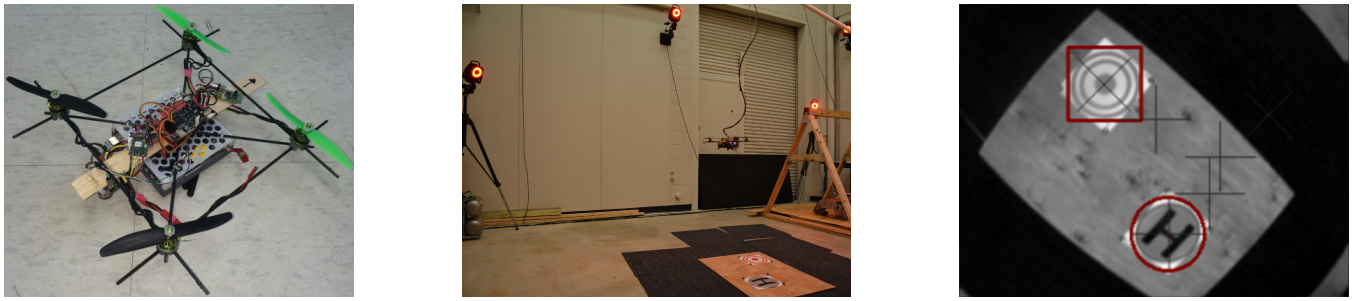
5. Flight Test

The flight tests were conducted using Georgia Tech UAV Simulation Tool (GUST)[7]. GUST is a software framework that the UAVRF has installed to many vehicles for developing and testing purposes. GUST provides hardware-in-the-loop (HITL), software-in-the-loop (SITL) and ground station software. The full state vector of the system in GUST for the indoor flight is expressed as follows:

$$\hat{x} = \begin{bmatrix} \hat{r} & \hat{p}_i & \hat{v}_i & {}^s\hat{b} & {}^\omega\hat{b} & {}^{fp}\hat{p}_1 & \dots & {}^{fp}\hat{p}_N \end{bmatrix}^T, \quad (57)$$

where N is the number of feature states, and ${}^{fp}\hat{p}_i$ denotes the position of the i-th feature point. The feature points are extracted using a corner Harris algorithm [5], and the feature

[7] **http://www.uavrf.gatech.edu/platforms/gust/**



(a) (b) (c)

Figure 8. GTQ mini, 500-gram MAV is shown in (a). The figure (b) shows GTQ mini flying in Georgia Tech Indoor Flight Facility. GTQ Mini is in flight with the safety-wire. The figure (c) shows the onboard image recorded in this experiment. The rectangle is the output of the target tracker that searches the double-red circle, and the circle is the one from the helipad finder.

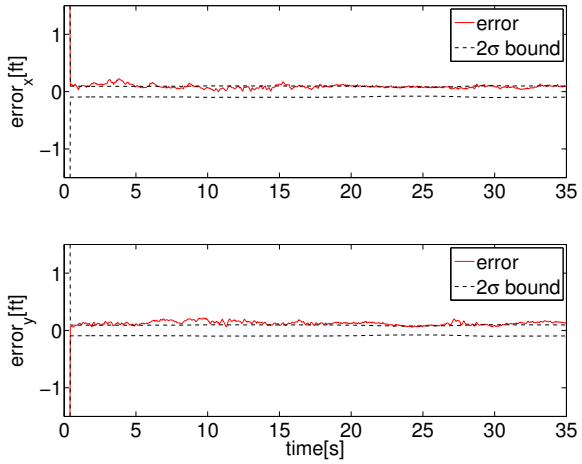


Figure 9. The plots show the error and two sigma bound of the double-red circle searching result after tuning. The errors are computed using the Vicon outputs. The position estimation is slightly biased to the positive direction.

states are assumed to be static. Instead of the conventional EKF used for the target tracking and simulation as described in the Section 2 and 3, navigation uses a Bierman-Thornton EKF. The BTEKF [4] uses modified Cholesky factors U and D on the covariance matrix P:

$$P = UDU^T, \tag{58}$$

where U is upper triangular matrix with a unit diagonal, and D is a diagonal matrix. The conventional EKF propagate (Equation (9)) is replaced by

$$U_k^- = L^T, D_k^- = G^T \begin{bmatrix} D_{k-1}^+ & 0 \\ 0 & Q_{k-1} \end{bmatrix} G, \tag{59}$$

where L is expressed in the equation below:

$$\begin{bmatrix} U_{k-1}^+ {}^T F_{k-1}^T \\ I \end{bmatrix} = GL. \tag{60}$$

The measurement update equation (12) is also expressed using U and D as follows:

$$U^+ = U^- U_a, \tag{61}$$

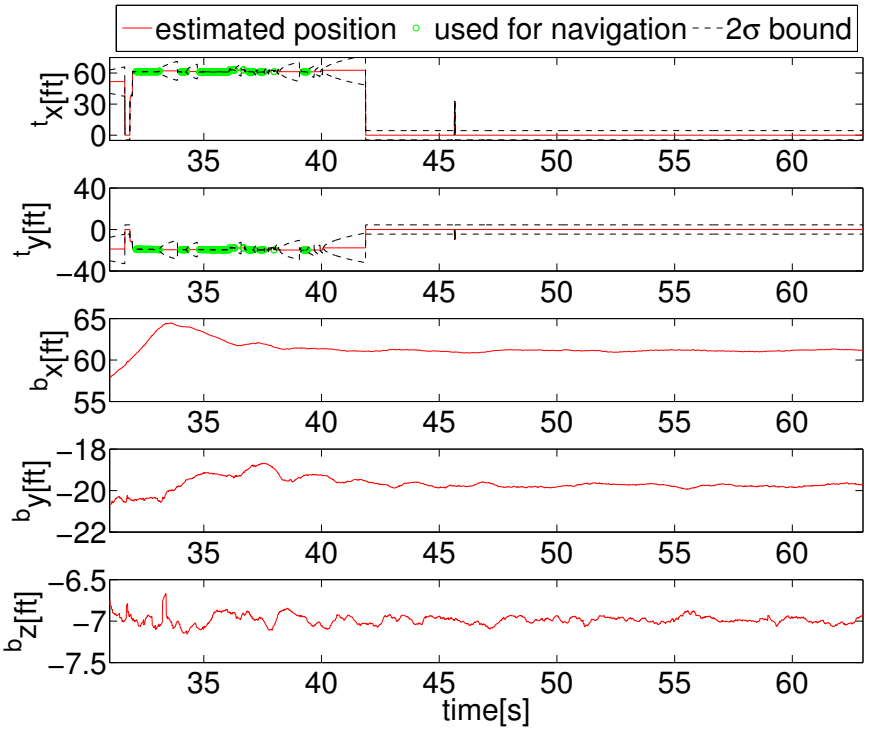
where

$$D^- - \frac{D^- U^{-T} H^T H U^- D^-}{H U^- D^- U^{-T} H^T + R} = U_a D^+ U_a^T. \tag{62}$$

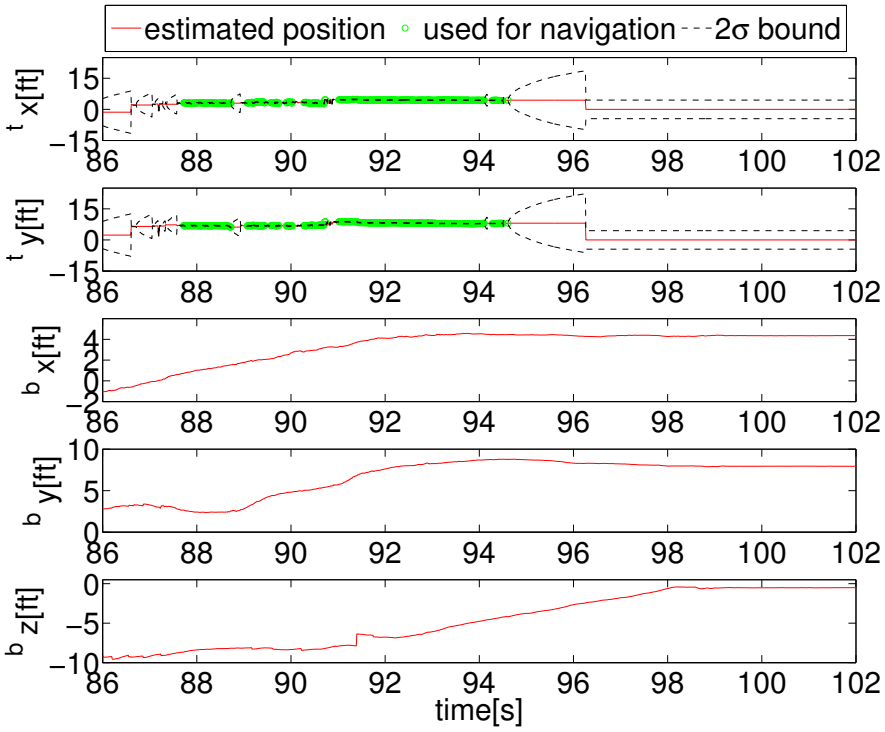
The details of the vision-aided inertial navigation is described here [15]. It is reported in [6] that the numerical stability is improved using this method.

Results from the AHS MAV Challenge are shown in Figure 10. The competition was held at the convention center in Virginia Beach with a snapshot of GTQ-mini flying seen in Figure 11. Figure 10 (a) shows that the target was found at 32[s]. Once the EKFs converged the onboard planner put a new waypoint at the specified target location where the vehicle completed a thirty second hover. The target was last seen at 40[s], and the estimator covariance diverged after this. One reason for this would be the shadow of the vehicle caused vision system to lose the target, and this explanation is supported by Figure 12 (a)-(c). As shown in the figures, the shadow of the vehicle is projected to the board when the vehicle is over the target. Since the machine-learning training was done with the images with no shadow, the Haar-like feature detector could not detect the target with shadows. In addition, the competition place was not as well-lit, and the target was relatively whited out due to the mis-tuned exposure parameter of the camera. This made the detection more challenging. However, the combination of Haar-like feature detector and multiple EKFs allowed the vehicle to hover over the desired point even after there was no new measurements. The landing was also accomplished as shown in the Figure 10, and the video of the full mission is available.[8] As were in the simulations presented in Section 4, the estimator lost the confidence after the vehicle descended to around 5ft, and stopped updating the desired waypoint. Both of the phases were successful, and the Georgia Aerial Robotics (GTAR) Team won first prize in the competition.[9]

[8] **https://youtu.be/LLHToSvsHgE**
[9] **https://vtol.org/news/winners-of-ahs-international-3rd-annual-mav-student-challenge**



(a)The result of the vision-based hovering. At 32 seconds the target was seen. At 64 seconds, the vehicle completed a thirty second hover.



(b)The result of the vision-based landing. At 87 seconds the helipad was seen. The vehicle successfully landed on the helipad at 98 seconds.

Figure 10. The flight result from the 2015 AHS MAV Student Challenge. This shows the estimated target position with two sigma bound, and the vehicle 3D position. The green dots are the points that satisfy the convergence criteria discussed in Section 2.

Figure 11. The convention center, Virginia Beach, VA. GTQ Mini in flight for practice.

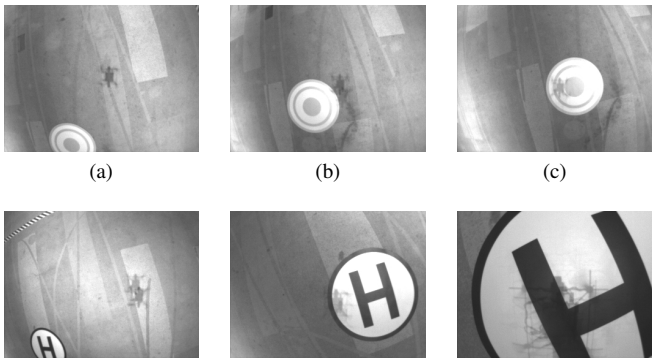


Figure 12. The onboard images recorded in hovering and landing. The target was found at (a), and the vehicle moved toward the target (b). When the vehicle was over the target (c), the shadow of the vehicle was projected to the target. The helipad was found at (d), and the vehicle attempted landing (e). As the vehicle descended, a part of the helipad was cut off (f), and the vision system got no new measurements after this.

6. Conclusion

This paper described a new target tracking method for MAVs using Haar-like feature detector and multiple EKFs. We developed a high performance MAV quadrotor vehicle capable of indoor navigation and target searching with no external aides. Both of the simulation and experiment results validate the suggested methodology. Also, flight test data is reported from the result of 2015 AHS MAV Student challenge. The accomplishment of the mission indicates that this technique is suited for search missions using MAVs.

Acknowledgments

The authors would like to thank all of the Georgia Tech Aerial Robotics(GTAR) and Unmanned Flying Club team members who made this development happen.

References

[1] Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. Journal of Field Robotics, 30(3), 415-438.

[2] Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11-15.

[3] Freund, Y., and Schapire, R. Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148-156, 1996.

[4] Grewal, M. S., & Andrews, A. P. (2014). Kalman filtering: Theory and Practice with MATLAB. John Wiley & Sons.

[5] Harris, C., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, p. 50)

[6] Haviland, S., Bershadsky, D., Magree, D. and Johnson, E., Development of a 500 gram Vision-based Autonomous Quadrotor Vehicle Capable of Indoor Navigation Proceedings of the 71st AHS Annual Forum, Virginia Beach, VA, May 5-7, 2015.

[7] Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Fraundorfer, F., & Pollefeys, M. (2014). Autonomous visual mapping and exploration with a micro aerial vehicle. Journal of Field Robotics, 31(4), 654-675.

[8] Horn, B. K., & Schunck, B. G. (1981, November). Determining optical flow. In 1981 Technical symposium east (pp. 319-331). International Society for Optics and Photonics.

[9] Jones, E. S., & Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. The International Journal of Robotics Research, 30(4), 407-430.

[10] Kelly, J., & Sukhatme, G. S. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. The International Journal of Robotics Research, 30(1), 56-79.

[11] Lefferts, E., Markley, F., and Shuster, M., Kalman filtering for spacecraft attitude estimation, AIAA Journal of Guidance, Control, and Dynamics, vol. 5, pp. 417-429, Sep-Oct 1982.

[12] Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on (Vol. 1, pp. I-900). IEEE.

[13] Lowe, D., Object recognition from local scale-invariant features, in Computer Vision, 1999. The proceedings of the Seventh IEEE International Conference on, vol. 2, pp. 1150-1157 vol2, 1999.

[14] Lucas, B. D., & Kanade, T. (1981, August). An iterative image registration technique with an application to stereo vision. In IJCAI (Vol. 81, pp. 674-679).

[15] Magree, D., & Johnson, E. N. (2015, January). A Monocular Vision-aided Inertial Navigation System with Improved Numerical Stability. In Proceedings of the AIAA Guidance Navigation and Control Conference, Kissimmee, Florida (pp. 6-2015).

[16] Masselli, A., Yang, S., Wenzel, K. E., & Zell, A. (2014). A cross-platform comparison of visual marker based ap-

proaches for autonomous flight of quadrocopters. Journal of Intelligent Robotic Systems, 73(1-4), 349-359.

[17] Matas, J., Galambos, C., & Kittler, J. (2000). Robust detection of lines using the progressive probabilistic hough transform. Computer Vision and Image Understanding, 78(1), 119-137.

[18] Marchand, E., Spindler, F., & Chaumette, F. (2005). ViSP for visual servoing: a generic software platform with a wide class of robot control skills. Robotics & Automation Magazine, IEEE, 12(4), 40-52.

[19] Martin, P., Salan, E. (2010, August). Generalized Multiplicative Extended Kalman Filter for aided attitude and heading reference system. In AIAA Guidance, Navigation, and Control Conference (p. 8300).

[20] Michael, N., Mellinger, D., Lindsey, Q., & Kumar, V. (2010). The grasp multiple micro-uav testbed. Robotics & Automation Magazine, IEEE, 17(3), 56-65.

[21] Schmid, K., Lutz, P., Tomi, T., Mair, E., & Hirschmller, H. (2014). Autonomous Visionbased Micro Air Vehicle for Indoor and Outdoor Navigation. Journal of Field Robotics, 31(4), 537-570.

[22] Shen, S., Mulgaonkar, Y., Michael, N., & Kumar, V. (2013, June). Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor. In Robotics: Science and Systems.

[23] Stevens, B. L., Lewis, F. L. (2003). Aircraft control and simulation. John Wiley Sons.

[24] Sung, K. K., & Poggio, T. (1998). Example-based learning for view-based human face detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(1), 39-51.

[25] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511). IEEE.

[26] Yang, S., Scherer, S. A., Zell, A. (2013). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. Journal of Intelligent Robotic Systems, 69(1-4), 499-515.

[27] Yang, S., Scherer, S. A., Schauwecker, K., Zell, A. (2014). Autonomous landing of MAVs on an arbitrarily textured landing site using onboard monocular vision. Journal of Intelligent Robotic Systems, 74(1-2), 27-43.

[28] Wu, A. D. (2010). Vision-based navigation and mapping for flight in GPS-denied environments.

BIOGRAPHY

Takuma Nakamura *is a PhD student working at the Georgia Tech UAVRF in Atlanta. He started his undergraduate study at Tohoku University, Sendai, Japan in 2009 and received a Bachelor of Science in Aerospace Engineering in March 2013. The following August, he joined the graduate study program in the School of Aerospace Engineering at Georgia Tech. He received a Master of Science in Aerospace Engineering from Georgia Tech in August 2015 and is continuing in the PhD program. His research interests include the computer vision systems, autonomous flight control for UAVs, 6degrees-of-freedom flight simulation and modeling. He is also a licensed private pilot, and hobby drone operator.*

Stephen Haviland *received his Aerospace Engineering B.S. from the University of Minnesota-Twin Cities summa cum laude in 2012. He was recently awarded his M.S. degree in Aerospace Engineering from the Georgia Institute of Technology in 2015 and is continuing on for his Ph.D. His research interests include modeling, simulation, vision-based navigation and control. On a personal level, he enjoys spending his free time flying remote control fixed-wing and multirotor vehicles.*

Dmitry Bershadsky *is a Ph.D. student working with at the Unmanned Aerial Vehicle Research Facility at Georgia Tech in Atlanta. He started his undergraduate study at Cornell University, Ithaca, NY in 2001 and received a Bachelor of Science in in Mechanical and Aerospace Engineering in 2004. He then received his Master of Engineering from the same university in 2005. For the next five years, he worked at Lockheed Martin Systems Integration in Owego, NY on avionics and algorithm development. In January 2010, he joined the graduate study program in the School of Aerospace Engineering at Georgia Tech. He received a Master of Science in Aerospace Engineering from Georgia Tech in 2013 and is continuing in the Ph.D. program.*

Daniel Magree *completed his undergraduate degree in Mathematical Science at Franciscan University of Steubenville. He earned a Masters degree in Aeronautical Engineering from the Air Force Institute of Technology, where he focused on aerospace control theory and computer vision applications. He completed his doctorate in Aerospace Engineering at Georgia Institute of Technology, researching vision-aided inertial navigation systems. His research interests include aerospace guidance, navigation and control, computer vision, and unmanned aerial vehicles.*

Eric N. Johnson *is the Lockheed Martin Associate Professor of Avionics Integration, School of Aerospace Engineering, Georgia Institute of Technology. He received a B.S. degree from University of Washington, M.S. degrees from MIT and The George Washington University, and a Ph.D. from Georgia Tech, all in Aerospace Engineering. He also has five years of industry experience working at Lockheed Martin and Draper Laboratory. He performs research in adaptive flight control, navigation, and autonomous systems.*